Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

A DATABASE FOR CATCH AND LENGTH COMPOSITION DATA RELATED TO THE SCOTIA-FUNDY LOBSTER FISHERY

C.M. Hunter and M.J. Tremblay

Biological Sciences Branch
Department of Fisheries and Oceans
Halifax, Nova Scotia B3J 2S7

JULY 1992

Canadian Manuscript Report of
Fisheries and Aquatic Sciences
No. 2149

Fisheries and Oceans Pêches et Océans

Canada

Canadian Manuscript Report of Fisheries and Aquatic Sciences

Manuscript reports contain scientific and technical information that contributes to existing knowledge but which deals with national or regional problems. Distribution is restricted to institutions or individuals located in particular regions of Canada. However, no restriction is placed on subject matter, and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Manuscript reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1–900 in this series were issued as Manuscript Reports (Biological Series) of the Biological Board of Canada, and subsequent to 1937 when the name of the Board was changed by Act of Parliament, as Manuscript Reports (Biological Series) of the Fisheries Research Board of Canada. Numbers 901–1425 were issued as Manuscript Reports of the Fisheries Research Board of Canada. Numbers 1426–1550 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Manuscript Reports. The current series name was changed with report number 1551.

Manuscript reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport manuscrit canadien des sciences halieutiques et aquatiques

Les rapports manuscrits contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui traitent de problèmes nationaux ou régionaux. La distribution en est limitée aux organismes et aux personnes de régions particulières du Canada. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports manuscrits peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports manuscrits sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annuel des publications scientifiques et techniques du Ministère.

Les numéros 1 à 900 de cette série ont été publiés à titre de manuscrits (série biologique) de l'Office de biologie du Canada, et après le changement de la désignation de cet organisme par décret du Parlement, en 1937, ont été classés comme manuscrits (série biologique) de l'Office des recherches sur les pêcheries du Canada. Les numéros 901 à 1425 ont été publiés à titre de rapports manuscrits de l'Office des recherches sur les pêcheries du Canada. Les numéros 1426 à 1550 sont parus à titre de rapports manuscrits du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 1551.

Les rapports manuscrits sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

Canadian Manuscript Report of
Fisheries and Aquatic Sciences 2149

1992

A DATABASE FOR CATCH AND LENGTH COMPOSITION DATA RELATED TO THE SCOTIA-FUNDY LOBSTER FISHERY

by

C. M. Hunter
and
M. J. Tremblay

Department of Fisheries and Oceans
Biological Sciences Branch
Halifax Research Laboratory
P. O. Box 550
Halifax, Nova Scotia
Canada B3J 2S7

© Minister of Supply and Services Canada 1992
Cat. No. Fs 97-4/2149E ISSN 0706-6473

Correct citation for this publication:

Hunter, C.M. and M. J. Tremblay. 1992. A database for catch and length composition data related to the Scotia-Fundy lobster fishery. Can. Manuscr. Rep. Fish. Aquat. Sci. 2149: 27 p.

93-00542/

TABLE OF CONTENTS

Abstract/Résumé	v
Introduction	1
ORACLE as the database management system	1
DATABASE MANAGEMENT AND PROGRAMMING NOTES	2
Accounts, updating and access	2
Links to a "backup account"	2
Programming "triggers"	2
Components of LOBDATA	2
Data Entry using LOGFORM or LENGFORM	5
Updating existing logbooks or existing length frequency samples	6
Data retrievals and queries using the application SQL*Plus	7
Starting an SQL*Plus session	7
Printing the queried results to a file for spooling	7
Library of Query command files	8
q1. List the logbooks by port and season, give the number of days fished	8
q2. As per q1 but for different LFA's	9
q3. Print summary of all logbook data by port and season	9
q4. As for q3 but for specified LFA	10
q5. Print summary of logbook data for specified period	11
q6. Print summary of logbook data for specified period and LFA	12
q7. Print summary of logbook data for specified intervals and port	12
q8. List the length frequency samples done at dockside (port) by port and season	14
q9. List the length frequency samples done at sea by port and season	14
q10. As for q8 but for specified LFA	15

(continued)

TABLE OF CONTENTS (continued)

q11. As for q9 but for specified LFA 15

q12. Give the relative length frequency distribution for a specifed port or sea sample 16

q13. Compute mean sizes of lobsters for a group of port or sea samples 17

Future Developments 18

Acknowledgements 18

References Cited 19

Figure 1. Lobster Fishing Areas in the Scotia-Fundy Region 20

Appendix 1. Programming notes on triggers 21

Appendix 2. SQL*Forms function keys for different terminal emulators 22

Appendix 3. Summary of Editing Operations and Keys 23

Hunter, C.M. and M. J. Tremblay. 1992. A database for catch and length composition data related to the Scotia-Fundy lobster fishery. Can. Manuscr. Rep. Fish. Aquat. Sci. 2149:27 p.

ABSTRACT

A database designed for the efficient storage and retrieval of data related to the Scotia-Fundy lobster fishery is described. The database ("LOBDATA"), utilizes ORACLE, a commercially available relational database management system. LOBDATA consists of a series of Tables for data storage. Closely related Tables are linked in "forms" for interactive data input. The forms allow input and update of (i) catch and effort data from volunteer logbooks, and (ii) length frequency data from samples of the fishing catches. To retrieve data summaries or data subsets, a set of Structured Query Language (SQL) command files has been developed. Users can customize these command files or create new ones as required. Data can be summarized and grouped by port, Lobster Fishing Area, or specified time period. In this document user features are emphasized, but aspects of interest to database programmers are also included.

Hunter, C.M. and M. J. Tremblay. 1992. A database for catch and length composition data related to the Scotia-Fundy lobster fishery. Can. Manuscr. Rep. Fish. Aquat. Sci. 2149:27 p.

RÉSUMÉ

On décrit ici une base de données conçue pour stocker et extraire efficacement des données sur la pêche du homard dans la région de Scotia-Fundy. Cette base de données («LOBDATA») fait appel à ORACLE, un système commercial de gestion de bases de données relationnelles. LOBDATA se compose d'une série de tables de stockage de données. Les tables qui sont étroitement associées les unes aux autres sont reliées sous forme de «grilles» servant à l'entrée interactive des données. Ces grilles permettent l'entrée et la mise à jour (i) de données sur les prises et l'effort de pêche émanant des journaux de pêche remis volontairement ainsi que (ii) de données sur la fréquence des longueurs provenant de l'échantillonnage des prises. On a conçu une série de fichiers de commande en langage relationnel pour extraire des sommaires ou des sous-ensembles de données. Les utilisateurs peuvent adapter ces fichiers à leurs besoins ou en créer de nouveaux. Les données peuvent être résumées et regroupées par port, par zone de pêche du homard, ou par période. Dans le présent document, on insiste sur les caractéristiques du système pour l'utilisateur, mais on aborde aussi certains éléments susceptibles d'intéresser les programmeurs de base de données.

INTRODUCTION

Biological advice for the assessment of the inshore lobster fishery in the Scotia-Fundy Region is based largely on three sources of data: logbooks kept voluntarily by fishermen, samples of the fishing catch (either at port or at sea), and landings statistics. The first two sets of data are collected by biologists in the Biological Sciences Branch, while landings statistics are collated by the Statistics Division, Program Coordination and Economics Branch. In the past, logbook and fishing catch data have been maintained by different biologists in a variety of forms including hard copies only (original data sheets), ASCII type data files, and files on spreadsheet type software (e.g. EXCEL).

A move to maintaining all the data in one database was desirable. There are 13 Lobster Fishing Areas (LFAs, Fig. 1) in the Scotia-Fundy Region, and these are assigned to different investigative units within the Biological Sciences Branch for the provision of biological advice. At present there are 5 such units in the Region (located in the Halifax Research Laboratory and the Biological Station in St. Andrews), and about 10 investigators in total. Given that the different LFAs are not biologically independent, and that responsibility for the different LFAs is fluid, it is important that lobster fishery data be readily accessible in a standardized form. A single database has several advantages: (i) current lobster investigators can examine data from adjacent areas if desired; (ii) less time necessary for responding to general data requests from client groups; (iii) outside users can access LOBDATA with assistance; and (iv) when the responsibility for an LFA changes, transitions should be smoother.

The report describes the structure and use of LOBDATA. It is directed at potential users of LOBDATA who are within the Scotia-Fundy region, and for others involved in lobster fisheries or in database management.

ORACLE AS THE DATABASE MANAGEMENT SYSTEM

ORACLE (a registered trademark of Oracle Corporation) is a relational database management system which uses an easy to learn language (Structured Query Language, or SQL) to query the database and write reports (Hursch and Hursch 1989; Pepin 1989). ORACLE is both versatile and powerful---it is available on a number of platforms (e.g. NOSVE on CYBER 860, VAX, IBM PCs and Macintosh), allows loading of data files in a number of formats, and can interface with a variety of languages such as PRO C and PRO FORTRAN. ORACLE has been adopted as the Department of Fisheries and Oceans standard for databases.

Two of the "tools" or applications of ORACLE (SQL*Plus and SQL*Forms, both trademarks of Oracle Corporation) have been used in the development of LOBDATA. SQL*Plus is the language for storing, manipulating, and reporting of data. In addition to standard SQL commands, SQL*Plus includes commands that allow users to customize reports and edit SQL statements. SQL*Forms is a facility for data entry and data updates. For further reference the reader is referred to Hursch and Hursch (1989), Pepin (1989), and the ORACLE User's Guides.

LOBDATA was developed primarily on an IBM PC linked to the Bedford Institute of Oceanography CYBER via the terminal emulator CONNECT. With a Macintosh linked to the CYBER via CONNECT, queries can be run, but data input using SQL*Forms is not possible. With a Macintosh configured with an appropriate Ethernet Card, NCSA Telnet Version 2.4 can be used as the terminal emulator, and SQL*Forms can be run from a Macintosh.

DATABASE MANAGEMENT AND PROGRAMMING NOTES

Accounts, updating and access

Any user with access to ORACLE will be able to sign onto LOBDATA and retrieve data. Changes or additions to LOBDATA will be restricted to those users with responsibilities for providing advice for one or more LFA. This security feature has been incorporated by linking port names to specific accounts.

Links to a "backup account"

All accounts will be linked to a backup account (currently C. Hunter's) upon signing on. Updates to the copy of LOBDATA on the backup account will be made only when users of authorized accounts issue an UPDATE command, or respond affirmatively to the query "save changes?".

Programming "triggers"

Triggers are sets of commands which are transparent to the user but are important to the database programmer. Triggers are begun by a certain event when a form is run. Each trigger may be composed of one or more steps, each of which contains one command. For further information refer to Appendix 1 and the SQL*Forms Designer's Reference.

COMPONENTS OF LOBDATA

LOBDATA consists of a series of Tables (LOGCODE, LOGDATA, LENGCODE, LENGDATA, LOCATIONS) developed in ORACLE's SQL*Plus. LOGCODE and LOGDATA are for data from fishing logbooks; LENGCODE and LENGDATA are for length frequency measurements taken at dockside or at sea. LOCATIONS is a table for grouping the data from individual ports by Lobster Fishing Area.

Using SQL*Forms, pairs of Tables have been grouped to make forms for data entry (indicated below). Each table contains a certain type of data. Data is entered primarily using the forms; data retrieval and querying is done in SQL*Plus. The contents of each table are shown below.

LOGCODE

Column	Data type	Field Description	Sample data
Port	Char(20)	Fishing port	SAMBRO
Fcode	Number(4)	Within-port code for fisherman	02
Season	Char(5)	Fishing season	88-89

LOGDATA

Column	Data type	Field Description	Sample data
Fcode	Number(4)	Within-port code for fisherman	02
Port	Char(20)	Fishing port	LOUISBOURG
Fdate	Date	Fishing date	01-JUN-88
N_trp	Number(4)	No. of traps	275
W_tot	Number(5)	Total weight (lbs)	375
Cpth	Number(6,2)	Catch-per-trap-haul	1.50

LOGDATA (cont'd)

Column	Data type	Field Description	Sample data
N_L	Number(3)	Number of lobsters[1]	100
W_avg	Number(5,2)	Average weight of lobsters[2]	1.25
NBf	Number(3)	No. of berried females[3]	3
W_C	Number(4)	Weight of canners[4]	100
CPTH_C	Number(5,2)	Catch-per-trap-haul: canners	0.40
SEASON	Char(5)	Fishing season	88-89

LENGCODE

Column	Data type	Field Description	Sample data
Port	Char(20)	Fishing port	SAMBRO
Sdate	Date	Sampling date	02-May-88
Season	Char(5)	Fishing season	88-89
Ntraps	Number(5)	Number of traps[5]	220
Latitude	Number(6)	Approximate latitude	463300
Longitude	Number(6)	Approximate longitude	625200

LENGFREQ

Column	Data type	Field Description	Sample data
Port	Char(20)	Fishing port	E. PASSAGE
Season	Char(5)	Fishing season	88-89
Sdate	Date	Sampling date	01-DEC-88
L_size	Number(3)	Lobster size (mm)	81
N_males	Number(4)	No. of males	3
N_fem	Number(4)	No. of females	1
Nbf	Number(4)	No. of berried females	0
Type	Char(1)	Port (P) or Sea (S) sample[6]	S

LOCATIONS

Column	Data type	Field Description	Sample data
LFA	Number(3)	Lobster fishing area no.	33
PORT	Char(20)	Fishing port	E. PASSAGE
COUNTY	Char(20)	County of fishing port	HALIFAX

SQL*Forms is a screen painter that allows for easier data entry, updates, and deletions. The Logbook form (called "LOGFORM") is a screen layout of blank spaces that is divided into 2 blocks, with each block representing 1 table. One is the logbook master table (LOGCODE),

---

[1] A few logbooks provide this information rather than total weight

[2] Where lobster number is given, an average individual lobster weight is used to estimate total weight

[3] This is an optional data item provided only by a few fishermen.

[4] Canner sized lobsters (< 81 mm carapace length) are fished only in LFA 27 in Scotia-Fundy

[5] This field and the next 2 would normally be completed only for sea samples.

[6] This field automatically becomes 'S' if the latitude field is filled out.

which has one record for each logbook. The other table (LOGDATA) contains the detailed records for each logbook. The form appears on the screen as follows:

LOG CODES

PORT	SAMBRO
FCODE	2
SEASON	88-89

LOG DATA

Code	Port	Fdate	N_trp	W_tot	Cpth	N_L	W_avg	Nbf	W_C	CPTH_C
2	SAMBRO	01-DEC-88	250	375	1.50					
2	SAMBRO	03-DEC-88	250	320	1.28					
2	SAMBRO	04-DEC-88	250	300	1.2					
2	SAMBRO	05-DEC-88	220	190	.86					
2	SAMBRO	06-DEC-88	250	220	.88					
2	SAMBRO	07-DEC-88	250	230	.92					
2	SAMBRO	01-JAN-89	250	120	.48					

In a similar manner, LENGCODE and LENGFREQ are combined in a form ("LENGFORM") for data entry. The form appears on the screen as follows:

LENGTH CODE

PORT	E. PASSAGE	**LATITUDE**
SDATE	01-DEC-88	**LONGITUDE**
SEASON	88-89	**NTRAPS**

LENGTH FREQUENCY DATA

Port	Season	Type	Sdate	L_size	N_males	N_fem	Nbf
E. PASSAGE	88-89	P	01-DEC-88	80	0	2	0
E. PASSAGE	88-89	P	01-DEC-88	81	3	2	0
E. PASSAGE	88-89	P	01-DEC-88	82	6	6	0
E. PASSAGE	88-89	P	01-DEC-88	83	3	3	0
E. PASSAGE	88-89	P	01-DEC-88	84	2	7	0
E. PASSAGE	88-89	P	01-DEC-88	85	9	6	0
E. PASSAGE	88-89	P	01-DEC-88	86	4	2	0

DATA ENTRY USING LOGFORM OR LENGFORM

Features

LOGFORM provides structured support for the entry of data . Some edit checks are performed automatically. For example when a date is entered, SQL*Forms has a built in calendar that checks all dates including leap years. SQL*Forms also reads the ORACLE data dictionary to ensure the information entered is of the correct data type. For example if a field was specified as numeric, and the user typed a character the following message would appear on the bottom of the screen: **Legal characters are 0-9.**

Other features of LOGFORM are as follows:

(1) calculates catch-per-trap-haul (CPTH) automatically;

(2) increments the date by 1 day automatically;

(3) requires input of port, fcode, and season;

(4) inserts the Port and userid into the correct table to ensure data security for each user.

The length frequency form (LENGFORM) is structured similarly to LOGFORM. LENGFORM is also divided into 2 blocks, with each block representing 1 table. One is the Length frequency sample master table (LENGCODE), which has one record for each length frequency sample. The other table (LENGFREQ) contains the detailed number at size data for each sample. Samples can be those done at dockside (Port samples) or those done at sea (Sea samples). Features of LENGFORM include:

(1) automatically increments size of lobsters by 1 mm (user inputs minimum length);

(2) automatically inserts zeroes into number of males, females, and berried fields (user must put all values only into first record i.e. number at minimum size);

(3) inserts the Port and userid into the correct table to ensure data security for each user;

(4 automatically inserts "P" for Port or "S" for sea samples depending on whether data is entered in the latitude field of the upper block.

Getting started in SQL*Forms

To enter LOGFORM type the following while in NOSVE:

RUNFORM LOGFORM username/password <cr>

To enter LENGFORM type:

RUNFORM LENGFORM username/password <cr>

Operations within Forms are controlled by hitting keys corresponding to certain functions. The functions are constant from system to system and are highlighted in **boldface** in the following text. The key (s) corresponding to a given function will depend upon the terminal emulator and keyboard definition; examples are given in Appendix 2. Below the keys corresponding to the boldface functions are given in brackets for an IBM PC using the terminal emulator program CONNECT.

When the user first enters a form it is in the active or normal mode. For entering new records the operator hits the **NEXT FIELD** (Tab or Return) key. Notice that when the cursor moves from the upper block (Log Codes in LOGFORM) to the lower block (Log Data in LOGFORM) values from the upper block are copied to the lower block, and will be part of each record within the lower block.

When the user has finished entering the detailed records for a specific logbook the **COMMIT** (End) key must be pressed to commit the records to the database. The information on the screen does not enter the database until a **COMMIT** is executed. Note that if an exit without committing the changes is desired, hit the **EXIT/CANCEL** (F3) key. With this function the option to quit without saving changes is available.

To go back a field press the **PREVIOUS FIELD** (Shift-Tab) key.

Several records can be entered at one time. The user can move up and down between rows, (**NEXT RECORD** and **PREVIOUS RECORD**; Down and Up arrow keys) and back and forth within rows.

To exit SQL*Forms hit the **EXIT/CANCEL** (F3) key.

Updating existing logbooks or existing length frequency samples

LOGFORM and LENGFORM are very useful for adding or making changes to records (i.e. detailed records) pertaining to an existing logbook or length frequency sample. First the user must query the logbook to the screen within LOGFORM or LENGFORM. This is done by pressing the **ENTER QUERY** (F1) key, and filling in the fields of the upper block to match that of the desired logbook or length frequency sample. This is followed by pressing the **EXECUTE QUERY** (F2) key, which will bring the existing logbook or length frequency data into the second block.

The second block is then entered by pressing the **NEXT FIELD** (tab or return) key. Move through the records by pressing the **NEXT SET OF RECORDS** (Shift F2) key until the row of interest is retrieved. Existing data can now be modified, or new data can be entered.

****NOTE**** - In order that unwanted changes are not made, caution must be used when updating because LOGFORM and LENGFORM are structured to do certain things automatically. For example if a user is at the end of a record in LOGFORM and the **NEXT FIELD** (Tab or Return) key is struck, the date from the current record is copied to the next record (actually current date plus one day). This will overwrite existing data, and if dates are not consecutive, the new date will be incorrect. In LENGFORM, 0's are automatically written to the frequency columns for the next length interval. These problems can be remedied in both LOGFORM and LENGFORM by using the **NEXT RECORD** (down arrow key) when at the end of a record instead of the **NEXT FIELD** (Tab or Return) key. When the changes are made commit the changes to the database using the **COMMIT** (End) key. Any insertions, updates or deletions made to the workspace do not affect the database until the **COMMIT** (End) key is pressed.

To insert a record on the form in an existing logbook or length frequency sample press the **CREATE RECORD** (F9) key. This opens up a blank space for each field on the screen for typing in the new record. To delete a record from a logbook, query the record to the screen. Move the cursor to the record you wish to delete by using **NEXT RECORD** (Down arrow) and **NEXT FIELD** (Tab or return) key. Press the **DELETE RECORD** (Shift-F5). Commit

the changes to the database using the **COMMIT** (End) key. Remember that only certain users can update or delete records.

If the user wishes to see which port codes were entered into the database, press the **LIST FIELD VALUES** (F4) key. Users can see the ports by pressing the **NEXT FIELD** (Tab or Return) key.

DATA RETRIEVALS AND QUERIES USING THE APPLICATION SQL*Plus

Starting an SQL*Plus session

SQL*Plus allows the user to retrieve information from the LOBDATA database. To start an SQL*Plus session type:

SQLPLUS username/password <cr>

To exit an SQL*Plus session type QUIT.

Data can be selected, ordered, and manipulated to some extent using the language of SQL. An example query using SQL is shown below.

```
  1  select port,fcode,fdate,season,avg(cpth),
  2  round(stddev(w_tot),1) "SD CPTH"
  3  from logdata
  4* group by port,fcode,fdate,season;
```

```
PORT                 FCODE FDATE    SEASON  AVG(CPTH)    SD CPTH
-------------------- ----------- --------- ----- ---------- ----------
ABBOTS HBR.                1 30-NOV-88 88-89       3.47          0
ABBOTS HBR.                1 01-DEC-88 88-89          4          0
ABBOTS HBR.                1 02-DEC-88 88-89       3.08          0
ABBOTS HBR.                1 03-DEC-88 88-89       2.77          0
ABBOTS HBR.                1 05-DEC-88 88-89       1.69          0
ABBOTS HBR.                1 06-DEC-88 88-89       1.67          0
.....................
.....................
BICKERTON                  1 20-APR-90 90-90       1.72          0
BICKERTON                  1 21-APR-90 90-90        .98          0
BICKERTON                  1 23-APR-90 90-90       1.04          0
BICKERTON                  1 24-APR-90 90-90       1.38          0
.....................
.....................
9158 records selected.
```

Note that each query must end with a semicolon.

Printing the queried results to a file for spooling

To spool the results of a query from within SQL*Plus, requires only 1 additional line in the command file. An example follows:

```
SQL>  Spool fn_lst                    fn_lst is a file name specified by the user
SQL>  Select * from Logcode;
SQL>  Spool off
```

SQL> **quit**

The user can now download the spooled file (fn_lst) to the Macintosh using a terminal emulation package.

LIBRARY OF QUERY COMMAND FILES

Many of the queries will be standard for all investigators (e.g. give statistics for CPTH for individual logbooks within a given port over a given time period), and thus a library of command files for common queries has been developed. These command files differ from the previous SQL commands in that they include ORACLE'S additional commands for formatting output (eg. break, btitle, column, compute, describe, edit, save, start, ttitle). These commands are not stored in the SQL buffer and do not need a semicolon (;) at the end of each command. All queries can be copied to any user account upon request.

To retrieve and list a command file type GET filename. To execute a command file (with or without GETing it first) type START filename.

Modification of a query command file can be done using whatever editor is available. The editor function key map will depend upon what editor and terminal emulator is in use. The edit keys for a Macintosh using Telnet as a terminal emulator, and for a PC using CONNECT as a terminal emulator are shown in Appendix 3.

What follows is a list of 13 query command files and some example output. Queries with an asterix require user modification to specify the area, time window, or other variable of interest. That part of the text requiring modification is written in **boldface**.

q1 List the logbooks by port and season give number of days fished.

```
SQL> get q1
  1  spool qlout
  2  set pagesize 65
  3  column season heading season
  4  column season format a10
  5  select port,season,count(distinct fcode) "N of log books"
  6  ,count(port) "N of days fished"
  7  from logdata
  8  group by port,season;
  9* spool off

SQL> start q1

PORT                 season     N of log books N of days fished
-------------------- ---------- -------------- ----------------
ABBOTS HBR.          88-89                   1               18
BICKERTON            90-90                   5              289
CANSO                90-90                   4              180
CLAM BAY             90-90                   2              105
E. PASSAGE           84-85                   2              112
GABERUS              91-91                   1               44
.....
.....
.....
65 records selected.
```

q2* As per q1 but for different LFA's.

```
SQL> get q2
  1  spool q2out
  2  set pagesize 65
  3  column season heading SEASON
  4  column season format a10
  5  select logdata.port,logdata.season,count(distinct fcode) "N of log
books"
  6  ,count(logdata.port) "N of days fished"
  7  from locations,logdata
  8  where upper(locations.port)=upper(logdata.port)
  9  and lfa=34
 10  group by logdata.port,logdata.season;
 11* spool off

SQL> start q2

PORT                 SEASON     N of log books N of days fished
-------------------- ---------- -------------- ----------------
PT. MAITLAND         88-89                  17             1228
PT. MAITLAND         89-90                  16             1189
.....................
.....................
.....................
```

q3 Print summary of all logbook data by port and season.

```
SQL> get q3
  1  spool q3out
  2  set pagesize 65
  3  ttitle 'Summary of logbook data'
  4  column season heading SEASON
  5  column season format a6
  6  column port format a15
  7  column fcode format 99999
  8  column sum(w_tot) heading 'TOT lbs'
  9  column avg(cpth) heading "MEAN CPTH"
 10  column round(stddev(cpth),2) heading 'SD CPTH'
 11  column sum(w_tot) format 9999999
 12  column avg(cpth) format 999.99
 13  select  port,season,fcode,sum(n_trp),sum(w_tot),avg(cpth),
 14  round(stddev(cpth),2)
 15  from logdata
 16  group by port,season,fcode;
 17* spool off

SQL> start q3

Tue Oct 22                                                    page    1
                          Summary of logbook data

PORT            SEASON  FCODE SUM(N_TRP)  TOT lbs MEAN CPTH    SD CPTH
--------------- ------ ------ ---------- -------- --------- ----------
ABBOTS HBR.     88-89       1       5260     8615      1.51       1.08
BICKERTON       90-90       1      10455     8674       .78        .41
```

```
BICKERTON       90-90       2      10958      7115       .62        .23
BICKERTON       90-90       3       3800      2533       .65        .29
BICKERTON       90-90       4      16139     10769       .71        .31
.....................
.....................
.....................
FOURCHU         90-90       1      14659     24505      1.67        .49
FOURCHU         90-90       2      12041     22414      1.85        .74
FOURCHU         91-91       1      14503     23690      1.63         .4
FOURCHU         91-91       2      12569     25662      2.04        .76
GABERUS         90-90       1      12003     13669      1.12        .43
GABERUS         91-91       1      11809     14451      1.21        .39
GLACE BAY       90-90       1      13700     10647       .78         .6
.....................
.....................
.....................
160 records selected.
```

q4* As for q3 but for specified LFA.

```
SQL> get q4
  1  spool q4out
  2  set pagesize 65
  3  ttitle 'Summary of logbook data'
  4  column season heading SEASON
  5  column season format a6
  6  column port format a15
  7  column fcode format 99999
  8  column sum(w_tot) heading 'TOT lbs'
  9  column avg(cpth) heading 'MEAN CPTH'
 10  column round(stddev(cpth),2) heading 'SD CPTH'
 11  column sum(w_tot) format 9999999
 12  column avg(cpth) format 999.99
 13  select logdata.port,logdata.season,fcode,
 14  sum(n_trp),sum(w_tot),avg(cpth),
 15  round(stddev(cpth),2)
 16  from logdata, locations
 17  where upper(locations.port) = upper(logdata.port)
 18  and lfa=33
 19  group by logdata.port,season,fcode;
 20* spool off

SQL> start q4

Tue Oct 22                                                    page    1
                          Summary of logbook data

PORT            SEASON  FCODE SUM(N_TRP)  TOT lbs MEAN CPTH    SD CPTH
--------------- ------ ------ ---------- -------- --------- ----------
E. PASSAGE      84-85       2      16753    12160       .74        .63
E. PASSAGE      84-85       4       5612     1358       .21        .11
E. PASSAGE      85-86       2       6625     4721      1.47        .95
E. PASSAGE      86-87       3       4413     2528       .56        .17
E. PASSAGE      87-88       1       4825     4903       .97        .48
E. PASSAGE      88-89       1       5463     4593       .77        .51
E. PASSAGE      88-89       2       4514     5072      1.03        .75
MOUTON          84-85       1      13792    14311      1.02       1.13
```

```
MOUTON          84-85       7      14466     16093       1.13       1.14
MOUTON          85-86       1      17677     21058       1.17        .97
MOUTON          85-86       7      13063      9999        .75        .25
.....................
.....................
54 records selected.
```

q5* Print summary of logbook data for specified period.

```
SQL> get q5
  1  spool q5out
  2  set pagesize 65
  3   ttitle 'Summary of logbook data for period NOV-24-88 to 30-NOV-88'
  4  column season heading SEASON
  5  column season format a6
  6  column min(fdate) heading MIN(DATE)
  7  column min(fdate) format a9
  8  column port format a15
  9  column fcode format 99999
 10  column sum(w_tot) heading 'TOT lbs'
 11  column avg(cpth) heading 'MEAN CPTH'
 12  column round(stddev(cpth),2) heading 'SD CPTH'
 13  column sum(w_tot) format 9999999
 14  column avg(cpth) format 999.99
 15  select port,season,min(fdate),sum(n_trp),sum(w_tot),avg(cpth),
 16  round(stddev(cpth),2)
 17  from logdata
 18   where fdate between '24-NOV-88' and '01-DEC-88'
 19  group by port,season;
 20* spool off

SQL> start q5

Tue Oct 22                                                        page    1
          Summary of logbook data for period NOV-24-88 to 30-NOV-88

PORT            SEASON MIN(DATE) SUM(N_TRP)  TOT lbs MEAN CPTH    SD CPTH
--------------- ------ --------- ---------- -------- --------- ----------
ABBOTS HBR.     88-89  30-NOV-88        750     2800      3.74        .37
E. PASSAGE      88-89  29-NOV-88       1446     2657      1.81        .75
LATOUR          88-89  29-NOV-88       1250     2871      2.30        .53
METEGHAN        88-89  30-NOV-88        450     2300      4.36       1.92
MOUTON          88-89  29-NOV-88       2241     4900      2.19        .88
PEMBROKE        88-89  30-NOV-88        752     5483      7.29        .37
WEST HEAD       88-89  30-NOV-88       1106     2300      2.09        .91
WHITE HD. ISL.  88-89  29-NOV-88        310     1100      3.55          0
WOODS HBR.      88-89  30-NOV-88        800     4300      5.38        .18
.....................
.....................
.....................
17 records selected.
```

q6* Print summary of logbook data for specified period and LFA.

```
SQL> get q6
  1  spool q6out
  2  set pagesize 65
  3   ttitle 'Summary of logbook data for period 24-NOV-88 to 01-DEC-88'
  4  column season heading SEASON
  5  column season format a6
  6  column min(fdate) heading MIN(DATE)
  7  column min(fdate) format a9
  8  column port format a15
  9  column fcode format 99999
 10  column sum(w_tot) heading 'TOT lbs'
 11  column avg(cpth) heading 'MEAN CPTH'
 12  column round(stddev(cpth),2) heading 'SD CPTH'
 13  column sum(w_tot) format 9999999
 14  column avg(cpth) format 999.99
 15  select logdata.port,logdata.season,min(fdate),sum(n_trp),
 16  sum(w_tot),avg(cpth),
 17  round(stddev(cpth),2)
 18  from logdata,locations
 19  where upper(locations.port) = upper(logdata.port)
 20  and lfa=33
 21   and fdate between '24-NOV-88' and '01-DEC-88'
 22  group by logdata.port,season;
 23* spool off

SQL> start q6

Tue Oct 22                                                        page    1
          Summary of logbook data for period 24-NOV-88 to 01-DEC-88

PORT            SEASON MIN(DATE) SUM(N_TRP)  TOT lbs MEAN CPTH    SD CPTH
--------------- ------ --------- ---------- -------- --------- ----------
E. PASSAGE      88-89  29-NOV-88       1446     2657      1.81        .75
MOUTON          88-89  29-NOV-88       2241     4900      2.19        .88
SAMBRO          88-89  29-NOV-88       1839     1923      1.47        .43
STONEHURST      88-89  29-NOV-88       2000     1733      1.61        .44
```

q7* Print summary of logbook data for specified intervals and port.

This complex query summarizes the CPTH data for intervals of 2 or more days over the course of one fishing season. The query requires two steps and there are thus two command files (q7a and q7). First the intervals are specified by creating a view (see ORACLE User's Guide) using q7a. A formula for obtaining various intervals is as follows:

trunc(fdate-startingdate +1)/Interval + (Interval-1/Interval))

To create the view for 7 day intervals, the formula must be modified and the first fishing day must be specified:

```
SQL> get q7a
  1  drop view vq7a;
  2  create view vq7a (port, season, fdate,fcode,int, n_trp,w_tot,cpth ) as
  3  select port,season,fdate,fcode,
```

```
  4  trunc((fdate - to_date('19-MAY-91')+1) /7 + (6/7)) ,
  5  n_trp,w_tot,cpth
  6* from logdata;
```

Now create the view by starting q7a:

```
SQL> start q7a

View dropped.

View created.
```

To generate the actual output, modify q7 accordingly and execute (start) it:

```
SQL> get q7
  1  spool q7out
  2  set pagesize 65
  3   ttitle 'Logbook summary: Fourchu, 1991 season'
  4  column fcode format 99
  5  column min(fdate) heading MIN(DATE)
  6  column min(fdate) format a9
  7  column port format a15
  8  column sum(w_tot) heading 'TOT lbs'
  9  column avg(cpth) heading 'AVG CPTH'
 10  column round(stddev(cpth),2) heading 'SD CPTH'
 11  column sum(n_trp) heading 'TOT TRAPS'
 12  column sum(n_trp) format 9999
 13  column sum(w_tot) format 99999
 14  column avg(cpth) format 99.99
 15  column int format 99
 16  select port,min(fdate),int,fcode,sum(n_trp),sum(w_tot),avg(cpth),
 17  round(stddev(cpth),2)
 18  from vq7a
 19   where fdate between '19-MAY-91' and '20-JUL-91' and
 20  int between 1 and 100 and
 21   port=upper('fourchu')
 22  group by fcode,port,int;
 23* spool off

SQL> start q7

Wed Dec 18                                                    page    1
                    Logbook summary: Fourchu, 1991 season
PORT            MIN(DATE)  INT FCODE TOT TRAPS TOT lbs AVG CPTH    SD CPTH
--------------- --------- --- ----- --------- ------- -------- ----------
FOURCHU         19-MAY-91   1     1      1749    3257     1.86        .65
FOURCHU         26-MAY-91   2     1      1750    3464     1.98        .46
FOURCHU         02-JUN-91   3     1      1750    3120     1.78        .35
FOURCHU         09-JUN-91   4     1      1640    2427     1.46         .4
FOURCHU         16-JUN-91   5     1      1750    3002     1.71        .17
FOURCHU         23-JUN-91   6     1      1493    2834     1.90        .34
FOURCHU         30-JUN-91   7     1      1500    2643     1.76        .16
FOURCHU         07-JUL-91   8     1      1128    1340     1.19        .36
FOURCHU         14-JUL-91   9     1      1743    3039     1.75        .27
FOURCHU         20-MAY-91   1     2      1485    4164     2.80       1.21
FOURCHU         27-MAY-91   2     2      1480    3697     2.50        .51
FOURCHU         03-JUN-91   3     2      1474    3203     2.17        .77
FOURCHU         10-JUN-91   4     2      1218    1828     1.50        .74
```

```
FOURCHU          17-JUN-91   5     2      1458     2573      1.77         .31
FOURCHU          24-JUN-91   6     2      1235     2734      2.22         .71
FOURCHU          01-JUL-91   7     2      1496     3014      2.01         .41
.....................
18 records selected.
```

q8 List the length frequency samples done at dockside (port) by port and season.

```
SQL> get q8
  1  spool q8out
  2  set pagesize 65
  3  ttitle 'List of port samples'
  4  column season heading Season
  5  column season format a10
  6  select port,season,sdate,sum(n_males) "No. Males",sum(n_fem) "No.
Females",
  7  sum(nbf) "No. Ovigerous"
  8  from lengfreq
  9  where type = upper('p')
 10  group by port,season,sdate;
 11* spool off

SQL> start q8

Thu Oct 17                                                    page    1
                          List of port samples

PORT             Season     SDATE      No. Males   No. Females No.Ovigerous
---------------- ---------- --------- ---------- ----------- ---------
INGONISH         91-91      10-JUL-91         114         224         0
LARDOISE         91-91      19-JUN-91         193         143         0
LOUISBOURG       91-91      03-JUN-91         145         202         0
PETIT DE GRAT    91-91      27-MAY-91         187         240         0
```

q9 List the length frequency samples done at sea by port and season.

```
SQL> get q9
  1  spool q9out
  2  set pagesize 65
  3  ttitle ' List of sea samples '
  4  column season heading Season
  5  column season format a10
  6  select port, season,sdate,sum(N_males) "No. Males",sum(n_fem)"No.
Females",
  7  sum(nbf) "No. Ovigerous"
  8  from lengfreq
  9  where type = upper('s')
 10  group by port,season,sdate;
 11* spool off
```

```
SQL> start q9

Thu Oct 17                                                    page    1
                          List of sea samples

PORT               Season     SDATE      No. Males No. Females No. Ovigerous
------------------ ---------- --------- ---------- ---------- -------------
FOURCHU            91-91      28-MAY-91        226        301            54
FOURCHU            91-91      16-JUL-91        223        121           145
GLACE BAY          91-91      20-JUN-91        503        410            67
NEILS HARBOUR      91-91      18-JUN-91        316        312            24
```

q10* As for q8 but for specified LFA.

```
SQL> get q10
  1  spool q10out
  2  set pagesize 65
  3  ttitle 'List of port samples for specified LFA'
  4  column season heading Season
  5  column season format a10
  6  select lengfreq.port,season,sdate,sum(n_males) "No. Males",
  7  sum(n_fem) "No. Females",sum(nbf) "No. Ovigerous"
  8  from lengfreq,locations
  9  where upper(locations.port) = upper(lengfreq.port)
 10  and type = upper('p')
 11  and lfa=27
 12  group by lengfreq.port,season,sdate;
 13* spool off

SQL> start q10

Mon Oct 28                                                    page    1
                  List of port samples for specified LFA

PORT               Season     SDATE      No. Males No. Females No. Ovigerous
------------------ ---------- --------- ---------- ---------- -------------
INGONISH           91-91      10-JUL-91        114        224             0
LOUISBOURG         91-91      03-JUN-91        145        202             0
```

q11* As for q9 but for specified LFA.

```
SQL> get q11
  1  spool q11out
  2  set pagesize 65
  3  ttitle 'List of sea samples for specified LFA'
  4  column season heading season
  5  column season format a10
  6  select lengfreq.port,season,sdate,sum(n_males) "No. Males",
  7  sum(n_fem) "No. Females",sum(nbf) "No. Ovigerous"
  8  from lengfreq,locations
  9  where type = upper('s')
 10  and upper(locations.port) = upper(lengfreq.port)
 11  and lfa=30
 12  group by lengfreq.port,season,sdate;
 13* spool off
```

```
SQL> start q11

Mon Oct 28                                                        page    1
                    List of sea samples for specified LFA

PORT                 season     SDATE      No. Males No. Females No. Ovigerous
-------------------- ---------- --------- ---------- ---------- -------------
FOURCHU              91-91      28-MAY-91        226        301            54
FOURCHU              91-91      16-JUL-91        223        121           145
.................................
.................................
```

q12* Give the relative length frequency distribution of lobsters for a specifed port or sea sample.

This query, like q7, requires two steps and there are two command files (q12a and q12). First a view is created with q12a:

```
SQL> get q12a
  1  drop view vq12a;
  2  create view vq12a (port, season, sdate, totmales,totfem, totberr) as
  3  select port,season,sdate,sum(N_males),sum(n_fem),sum(nbf)
  4  from lengfreq
  5* group by port,season,sdate;

SQL> start q12a

View dropped.


View created.
```

To generate the output, specify the port and date of the length frequency sample in the q12 command file and execute it:

```
SQL> get q12
  1  spool q12out
  2  set pagesize 65
  3  column port format a11
  4  column sdate format a9
  5  column L_size heading 'SIZE'
  6  column L_size format 999
  7  column %totm format 999.9
  8  column totn format 99999
  9  select lengfreq.port,lengfreq.sdate,L_size,n_males ,
 10  n_fem,nbf,(totmales+totfem+totberr) "TOTN",
 11  round((((n_males+n_fem+nbf)/(totmales+totfem+totberr+.001))*100),1)
 12  "%FREQ."
 13  from lengfreq,vq12a
 14  where lengfreq.port=vq12a.port and lengfreq.sdate=vq12a.sdate and
 15   lengfreq.port = upper('fourchu') and vq12a.sdate='28-MAY-91';
 16* spool off
```

```
SQL> start q12

PORT        SDATE      SIZE    N_MALES      N_FEM        NBF    TOTN     %FREQ.
----------- --------- ---- ---------- ---------- ---------- ------ ----------
FOURCHU     28-MAY-91   64          1          0          0    581         .2
FOURCHU     28-MAY-91   65          0          0          0    581          0
FOURCHU     28-MAY-91   66          0          1          0    581         .2
FOURCHU     28-MAY-91   67          1          0          0    581         .2
FOURCHU     28-MAY-91   68          2          1          0    581         .5
FOURCHU     28-MAY-91   69          2          0          0    581         .3
FOURCHU     28-MAY-91   70          3          1          0    581         .7
FOURCHU     28-MAY-91   71          0          1          0    581         .2
FOURCHU     28-MAY-91   72          3          7          0    581        1.7
FOURCHU     28-MAY-91   73          0          4          0    581         .7
FOURCHU     28-MAY-91   74          3          6          0    581        1.5
FOURCHU     28-MAY-91   75          5          1          2    581        1.4
FOURCHU     28-MAY-91   76          2          5          2    581        1.5
FOURCHU     28-MAY-91   77          8          9          1    581        3.1
FOURCHU     28-MAY-91   78          6          8          1    581        2.6
FOURCHU     28-MAY-91   79          4          5          0    581        1.5
FOURCHU     28-MAY-91   80          9          7          2    581        3.1
FOURCHU     28-MAY-91   81          7         12          3    581        3.8
.....
.....
.....
FOURCHU     28-MAY-91  103          6          1          1    581        1.4
FOURCHU     28-MAY-91  104          2          2          0    581         .7
FOURCHU     28-MAY-91  105          2          1          1    581         .7

85 records selected.
```

q13* Compute mean sizes of lobsters for a group of port or sea samples.

This query, also requires two steps and there are two command files (q13a and q13). First a view is created with q13a. Note that the port and time interval of interest must be specified.

```
SQL> get q13a
  1  drop view vq13a;
  2  create view vq13a (port, season, sdate,type, totmales,totfem, totberr,
  3                     totmalesize,totfemsize,totnbfsize ) as
  4  select port,season,sdate,type,sum(N_males),sum(n_fem),sum(nbf),
  5  sum(L_size*N_males),sum(L_size*N_fem),sum(L_size*nbf)
  6  from lengfreq
  7   where port=upper('Petit de grat') and sdate between '01-JAN-91' and
'31-DEC-91'
  8* group by port,season,sdate,type;
```

Now run it:

```
SQL> start q13a

View dropped.


View created.
```

Next get and run q13. Again specify the port and time interval:

```
SQL> get q13
  1  spool q13out
  2  set pagesize 65
  3  ttitle 'Mean sizes for Petit de Grat from 1991 port(p) or sea(s)
samples'
  4  column port format a14
  5  column sdate format a9
  6  column l_size heading 'SIZE'
  7  column L_size format 999
  8  column type heading 'Type'
  9  column type format a5
 10  column round((totmalesize)/(totmales+.001),1) format 999.9
 11  column round((totfemsize)/(totfem+.001),1) format 999.9
 12  column round((totnbfsize)/(totberr+.001),1) format 999.9
 13  select port,type,sdate,
 14  round((totmalesize)/(totmales+.001),1)  "Males",
 15  round((totfemsize)/(totfem+.001),1)  "Females",
 16  round((totnbfsize)/(totberr+.001),1)  "Berr",
 17  round((totmalesize+totfemsize+totnbfsize)/
(totmales+totfem+totberr+.001),1) "Tot"
 19  from vq13a
 20   where port = upper('petit da grat');
 21* spool off


SQL> start q13

Thu Dec 19                                                        page    1
        Mean sizes for Petit de Grat from 1991 port(p) or sea(s) samples

PORT           Type  SDATE          Males    Females       Berr        Tot
-------------- ----- --------- ---------- ---------- ---------- ----------
PETIT DE GRAT  P     27-MAY-91       94.1       91.6          0       92.7
PETIT DE GRAT  P     12-JUN-91       90.2       90.7          0       90.5
PETIT DE GRAT  S     13-JUN-91         75       79.6          0       77.1
PETIT DE GRAT  S     25-JUN-91       85.3       84.7       83.4       84.8
```

Future Developments

In the near future LOBDATA will be available on more platforms, including ORACLE on the VAX (at the St. Andrews Biological Station) and on individual microcomputers. Query development will be ongoing as users work more with the database. Additions will be made to the Locations table as data from more ports are added.

Future enhancements to LOBDATA may include facilities for interfacing with graphics software, and the integration of landings statistics (from DFO Statistics Division) into the database.

Acknowledgements

We wish to thank D. Pezzack, D. Duggan, R. Duggan, M. Eagles, and D. Swetnam for discussions on the contents and structure of the database, and for comments on the manuscript.

References Cited

Hursch, J. L. and C. J. Hursch. 1989. Working with ORACLE® Version 6.0. Windcrest Books, Blue Ridge Summit, PA.

Pepin, D. 1989. ORACLE® Programmer's Guide.QUE® Corporation, Carmel IN.

Oracle Corporation. 1988. SQL Language Reference Manual, Version 6.0. 1990. Oracle Corporation, Belmont, California.

Oracle Corporation. 1988. SQL*Forms® Designer's Reference, Version 2.0. 1988. Oracle Corporation, Belmont, California.

Oracle Corporation. 1988. SQL*Plus® User's Guide, Version 2.0. 1987. Oracle Corporation, Belmont, California.

Figure 1. Lobster fishing areas (LFAs) in the Scotia-Fundy Region as of 1991.

Appendix 1. Programming notes on triggers

Triggers have been incorporated into the forms to serve several functions. These include: (i) security for LOBDATA by associating port codes to specific user accounts; (ii) automatic calculations on data entered; (iii) ease of data entry by automatically incrementing certain fields with default data (0's) which can be overwritten by the user.

All triggers are transparent to the general user. As an example of these triggers for the database manager, the security triggers are explained below:

The Pre-update and pre-delete triggers are placed at the form level.

```
Select 'X' from dual
where :code.port in
(Select port_id from userid
where user_id=user)
```

where user is the userid given to a user upon login. Userid is the look-up table with a list of ports for each user.

At the block level(code block) a pre-insert trigger is created;

```
Insert into Userid(User_id,Port_id)
Values (user,:code.Port)
```

A key-nxtfld trigger allows the master detail relationship to take place.

```
#EXEMACRO NOFAIL nxtblk;nxtfld;nxtfld;
```

A key-nxtfld is placed on the W_tot field of the second block

```
#EXEMACRO NOFAIL CASE W-tot is
when '' then nxtfld;nxtfld;
when others then exetrg compute_wt;nxtfld;nxtfld;nxtfld;nxtfld;
endcase;
```

compute_wt trigger

```
Step 1:  Select to_number(:data.w_tot)/to_number(:data.n_trp)
         into :data.cpth
         from system.dual

Step 2:  Select round(:data.cpth,2)
         into :data.cpth
         from system.dual
```

Further details on triggers are available from C. Hunter.

Appendix 2. SQL*Forms function keys for different terminal emulators

Function	IBM PC with CONNECT Keystroke(s)	Macintosh with Telnet Keystroke(s) (no. on pad)
Block Menu	Ctrl-F1	Escape-b
Clear Block	Shift-F6	3
Clear Field	F6	9
Clear Form/Rollback	Home	Escape-c
Clear Record	F5	6
Commit Transaction	End	PF3
Count Query Hits	Shift-F1	Escape-4
Create Record	F9	PF2
Delete Backward	Backspace	DELETE
Delete Character	Del	
Delete Record	Shift-F5	Escape-d
Display Error	Shift-F10	,
Duplicate Field	Shift-F7	Escape-1
Duplicate Record	F7	Escape-2
Enter Query	F1	ENTER
Execute Query	F2	PF1
Exit/Cancel	F3	PF4
Help	F10	0
Insert/Replace	Ins	.
Left	Left arrow	Left arrow
List Field Values	F4	Escape-v
Next Block	Alt j[i]	1
Next Field	Tab or Return	
Next Primary Key Fld	Escape-Tab	Escape-Tab
Next Record	Down arrow	Down arrow or 4
Next Set of Records	Shift F2	Escape-s
Previous Block	Alt u[i]	2
Previous Field	Shift-Tab	8
Previous Record	Up arrow	Up arrow or 5
Print	Shift-F9	Escape-p
Redisplay Page	Shift-F8	-
Right	Right arrow	Right arrow
Scroll Left	Ctrl-Left arrow	Escape-Left arrow
Scroll Right	Ctrl-Right arrow	Escape-Right arrow
Show Function Keys	F8	Escape-k
Interrupt		Ctrl-Y

[i] if you are using an IBM PC and the Alt j and Alt u keys do not perform the desired function, define the keys by using the DEFINE option and specifiying:

alt j = 1b 4f 21 Od
alt u = 1b 4f 25 Od

The Define option is described in the Control Data CONNECT for the IBM PC Version 1.4 User's Guide.

Appendix 3. Summary of Editing Operations and Keys

Function	IBM PC with CONNECT Keystroke(s)	Macintosh with Telnet (VT100 emulation) Keystroke(s) (no. on pad)
First (beginning of file)	Ctrl-PgUp	0 followed by 1
Last (end of file)	Ctrl-PgDn	0 followed by 2
Middle (middle of sreen)	Ctrl-F1	Enter
Fwd (next screen of file)	PgDn	2
Bkw (previous screen of file)	PgUp	1
Quit (End and Save session)	F6	6
Exit (End seeion without saving)	Shift-F6	0 followed by 6
Back (Ends session, save changes &resume on 1st file)	F3	3
Undo	F5	5
Help	F4	4
Home (editor home line)	Home	0 followed by 3
Refrsh (rewrite screen)	Alt-I	0 followed by Enter
Width	Shift-F4	0 followed by PF4
Mark	F2	PF1
Unmrk	Shift-F2	0 followed by PF1
MrkCh	F7	PF2
MrkBx	Shift-F7	0 followed by PF2
Copy	F1	0 followed by 4
Move	Shift-F1	9
InsCh (Inserts a blank character)	Ins	7
InsLn (Inserts a blank line)	Alt-I	8
DelCh (Deletes current character)	Del	0 followed by 7
DelLn (Deletes current line)	Alt-D	0 followed by 8
Locate	F8	PF3
LocNxt	Shift-F8	0 followed by PF3
LocAll	Shift-F3	PF4
CirEl (deletes all characters to end of line)	Shift-F10	0 followed by ,(comma)
SkpEl (skip to end of current line)	F10	,(comma)
Join	Shift-F9	0 followed by - (minus sign)
Break	F9	- (minus sign)
LinUp (positions current line to top of the screen)	Ctrl-F2	0 followed by 9